RAIDIOS (RAID I/O Steering)

Design Guide, Version 1.2

August 2002

Order Number: 273724-003

Contents

Figures

Tables

1.0 Introduction

This document provides guidelines for implementing RAID I/O Steering (RAIDIOS), which is a specification that Intel® developed for motherboards and zero-channel RAID cards. RAIDIOS enables Modular RAID on Motherboard (MROMB) add-in adapters in targeted computer systems. RAIDIOS is also used in ROMB applications where the I/O processor (IOP) and the I/O controller are in a *single bus* architecture. See Section 1.4 for details on the single bus architecture.

Note: Use the information in this design guide as a guideline. Employ best-known design practices using board-level simulation, signal integrity testing and validation for a robust design.

Table 1. Revision History

Document	Comments
Version 1.0 Order# 273724-001	Initial Release
Version 1.1 Order# 273724-002	• Section 2.2.1—Updated implementation steps with MUX specific information. • Figure 5—Removed the pull-down resistor at IDSEL on the I/O controller. • Figure 6—Removed the pull-down resistor at IDSEL on the I/O controller. Replaced bus switch with MUX. • Section 2.3—Inserted new section "Multiple I/O Controller Considerations" • Section 4.3—Inserted new section "Impact to RAID Firmware of Multiple I/O Controllers" • Updated RAIDIOS schematic in appendix A with multiple I/O controllers

1.1 Reference Documents

MROMB Design Considerations Using the Intel® 80303 I/O Processor, Application Note, order No. 273593-001

RAID on Motherboard (ROMB) Considerations Using Intel 80321 I/O Processor, Application Note, order No. 273456-005

Additional information can be found on the Intel Developer Web Site at developer.intel.com/design/iio/

1.2 Document Organization

This document is divided into three sections and two appendices.

Section 2.0, "System Motherboard Design Considerations for Implementing RAIDIOS" provides theory of operation and implementation of each portion of the RAIDIOS circuit.

Section 3.0, "RAIDIOS Design Specifications" provides the specifications that must be followed when selecting components for and laying out the RAIDIOS circuit.

Section 4.0, "MROMB Adapter Design Considerations for RAIDIOS" provides design considerations required to implement MROMB solutions using RAIDIOS.

Appendix A, "RAIDIOS Schematic" is a schematic diagram of the RAIDIOS circuit. Appendix B, "Timing Diagrams" are timing diagram examples that illustrate the PCI and PCI-X timing requirements for the RAIDIOS circuit.

1.3 Document Conventions

The following are definitions, explanations of terms, or formats used throughout this document.

An **I/O Controller** is any I/O device used as the interface to storage devices, for example, SCSI, Fibre Channel, S-ATA, etc. The term I/O controller is used in this document to represent any I/O disk controllers or translators.

MROMB, also referred to as Zero-Channel RAID (ZCR), is a cost effective RAID hardware solution for motherboards. It allows motherboard vendors to have a hardware-based RAID subsystem without an IOP embedded on the motherboard.

Motherboard, **systemboard**, and **mainboard** are synonymous. They all refer to the main printed circuit board in the computer system.

Interrupt routing (also called I/O Steering) refers to that part of the RAIDIOS circuit that determines if the *system interrupt controller* or the *MROMB IOP* service interrupts intended for the I/O controller.

Control circuit (also called IDSEL control circuit) refers to the portion of the RAIDIOS circuit that hides and un-hides the I/O controller from the host. This consists of two parts: (1) a gate or switch on the motherboard to hide and un-hide the I/O controller's IDSEL signal, and (2) logic on the MROMB adapter to enable/disable the gate.

PCI and **PCI-X** refer to the system I/O bus interconnect technology. Note: Because RAIDIOS is independent of I/O bus technology, whenever PCI is referred to in the text or any figure in this document, it also refers to PCI-X unless otherwise indicated.

1.4 RAIDIOS, MROMB, and ROMB Overview

MROMB and ROMB are hardware RAID solutions that use an I/O controller that is integrated on the motherboard in conjunction with an IOP to form a complete hardware RAID subsystem. ROMB integrates the IOP onto the motherboard while MROMB gives the user the option of plugging the IOP into a PCI (or PCI-X) expansion slot. Additional circuitry is required on the motherboard to enable this capabilty for MROMB. That circuitry is RAIDIOS.

While RAIDIOS is used as the enabling circuit for MROMB, RAIDIOS is also required for the ROMB *single-bus* architecture solution because the system architecture for ROMB and MROMB are the same. ROMB can also be a *dual-bus* architecture depending upon the IOP used.

1.4.1 Single Bus Architecture

Single Bus architecture means that the IOP and the I/O controller are located on the same system bus segment. All MROMB solutions are single bus. The MROMB adapter, with its IOP, is inserted into an expansion slot that is located on the same system bus segment as the I/O controller. All communications between the IOP and the I/O controller must therefore be carried out over the system bus, visible to the host system. See Figure 1 for an overall view of this architecture as it applies to MROMB. ROMB single bus architecture is similar to MROMB except that the IOP and

the I/O controller are integrated directly onto the motherboard and located on the same system bus segment. ROMB single bus architecture is formed when the IOP does not have an integrated bridge (for example, the Intel® 80321 IOP). Figure 2 shows an example of this architecture.

Since both the IOP and the I/O controller are located on the same system bus segment in both the ROMB Single Bus and all MROMB solutions, both solutions require *interrupt routing* to route I/O interrupts and a *control circuit* that allows the IOP to configure and hide the I/O controller from the host. The control circuit does this by connecting and disconnecting the I/O controller's IDSEL (Initialization Device Select) signal from the PCI (or PCI-X) system bus.

1.4.2 Dual Bus Architecture

The other ROMB architecture is *dual-bus*. A bridge is integrated into the IOP (for example, the Intel® 80303 IOP) creating a secondary bus interconnect between the IOP and the I/O controller. The IOP is thus able to conduct all communications with the I/O controller via the secondary bus hidden behind the bridge. Figure 3 shows an example of this architecture. RAIDIOS is not required for this ROMB solution, but for the I/O controller to be used as a stand-alone I/O controller, RAIDIOS must re-route the I/O controller's IDSEL and interrupts. Thus RAIDIOS gives the platform designer greater design flexibility.

Figure 1. MROMB Architecture Block Diagram

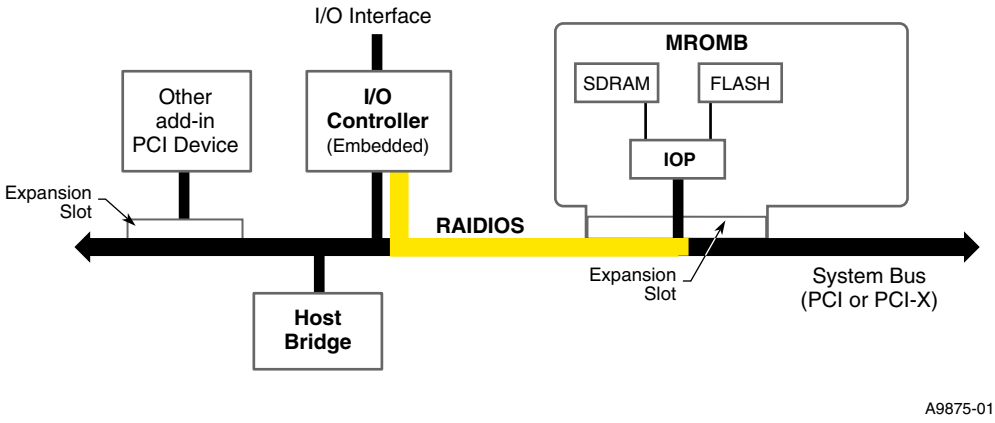
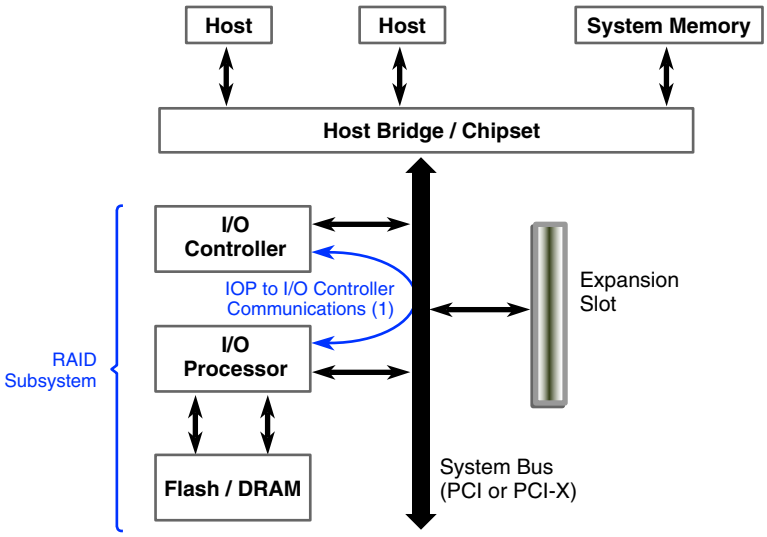


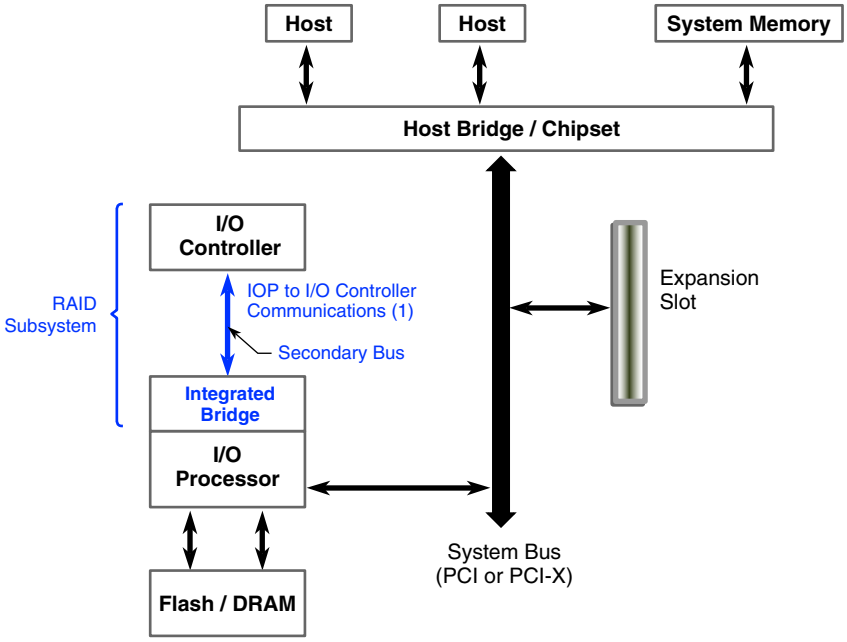
Figure 2. Single Bus ROMB System Architecture



(1) Note: Requires I/O interrupt routing and IDSEL control circuit, e.g., **RADIOS.**

A9876-01

Figure 3. Dual Bus ROMB System Architecture



(1) Note: All communications between the IOP and the I/O controller occur behind the bridge, hidden from the host.

A9877-01

2.0 System Motherboard Design Considerations for Implementing RAIDIOS

For an MROMB add-in adapter to function within a system motherboard, one PCI (or PCI-X) expansion slot must be modified with the RAIDIOS circuit. RAIDIOS is comprised of two functional components, the *interrupt routing circuit* and the *IDSEL control circuit*.

Interrupt routing refers to modifying the I/O controller's interrupt signal lines so that when the MROMB adapter is present, the I/O controller's interrupt signals are re-routed to the RAIDIOS expansion slot. This allows the MROMB's IOP to service interrupt requests intended for the I/O controller. When the MROMB adapter is not present, the I/O controller's interrupt lines are routed to the *system interrupt controller* so that it may service interrupt requests intended for the I/O controller.

The *IDSEL control circuit* allows the MROMB adapter to take control of and configure the I/O controller during PCI configuration cycles and then *hide* the I/O controller from the host at all other times. By inserting a switch in the I/O controller's IDSEL line between the I/O controller and the PCI (or PCI-X) bus, the MROMB adapter can hide the I/O controller from the host and use the I/O controller as part of its own RAID subsystem.

2.1 Routing the Embedded I/O Controller's Interrupts (INTx)

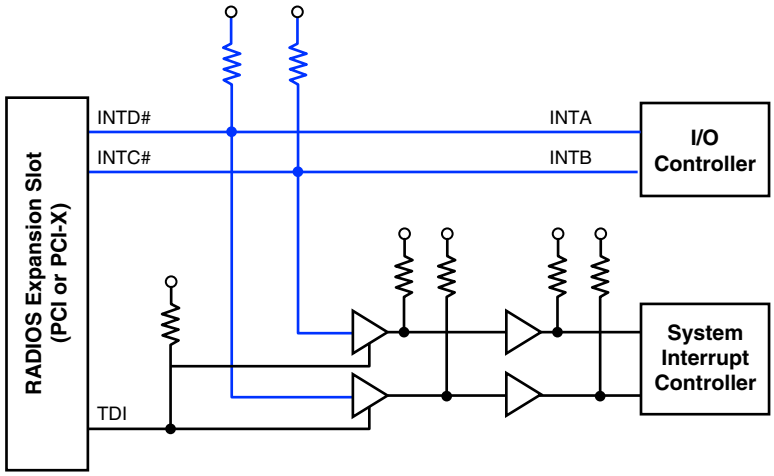
In the standard implementation of an embedded I/O controller, its interrupt lines are routed to the motherboard's system interrupt controller. An interrupt request sent from the I/O device's host driver is received by the system interrupt controller and serviced appropriately. RAIDIOS allows the MROMB adapter to service interrupts from the host driver (RAID driver) and forward them to the I/O controller as necessary.

2.1.1 Implementation

1. Route the I/O controller's interrupts to the expansion slot targeted for RAIDIOS (see Figure 4). Connect a signal line from INTD# on the RAIDIOS expansion slot to a point on the signal line that runs from INTA on the I/O controller to the system interrupt controller. Connect a signal line from INTC# on the RAIDIOS expansion slot to a point on the signal line that runs from INTB on the I/O controller to the system interrupt controller.
2. Between the branch points and the system interrupt controller, insert tri-state buffers and hex buffers. The hex buffers are included so that the I/O controller interrupts can be wire-OR shared with other devices. The interrupts are, by definition, open-collector so that the design engineer has the ability to wire-OR multiple devices on the same interrupt. The tri-state buffers that are shown in the RAIDIOS example schematic are active drive, both high and low, when they are enabled. This would prevent other devices from using these interrupt lines. Because the hex buffers are true open-collector devices, they give the system design engineer the ability to wire-OR multiple devices on the same interrupt lines. This implementation is PCI compliant.
3. Connect a signal line from *TDI* on the RAIDIOS expansion slot to the control (enable) inputs of the tri-state buffers.

A *pull-up resistor* is required on the TDI signal line to ensure that the signal does not float when the MROMB adapter is not inserted in the RAIDIOS expansion slot.

Figure 4. I/O Interrupt Routing Circuit Implementation



2.1.2 Theory of Operation

Inserting a MROMB adapter into the RAIDIOS-enabled slot and powering on the system causes the JTAG signal TDI to be pulled low (this is a requirement of the MROMB adapter designer, not the motherboard designer). When TDI is pulled low, the tri-state buffers, connected to TDI, are disabled. This re-routes the I/O controller's INTA and INTB signals away from the system interrupt controller to the RAIDIOS expansion slot. This allows the MROMB adapter's IOP to service the interrupt requests of the I/O controller. Removing the MROMB adapter enables the tri-state buffers (pull-up resistors keep this input from floating when the MROMB adapter is not present). Enabling the tri-state buffers returns the servicing of the I/O controller's interrupts to the system interrupt controller. Refer to Appendix A for a schematic view of this circuit.

2.2 Configuring and Controlling the Embedded I/O Controller

When the system boots after a reset, the host initiates a PCI bus scan to detect the PCI devices installed in the system, including the I/O controller. The host bridge issues PCI configuration cycles by asserting each AD[x] line (which are tied to IDSEL signal lines on the motherboard) to identify each PCI device and assign the necessary system resources. The MROMB adapter must also issue PCI configuration cycles for it to configure the I/O controller. If both the host and the MROMB adapter were allowed to configure the I/O controller it would lead to resource and interrupt conflicts. RAIDIOS prevents this from occurring with the *IDSEL control* portion of the RAIDIOS circuit.

2.2.1 Implementation

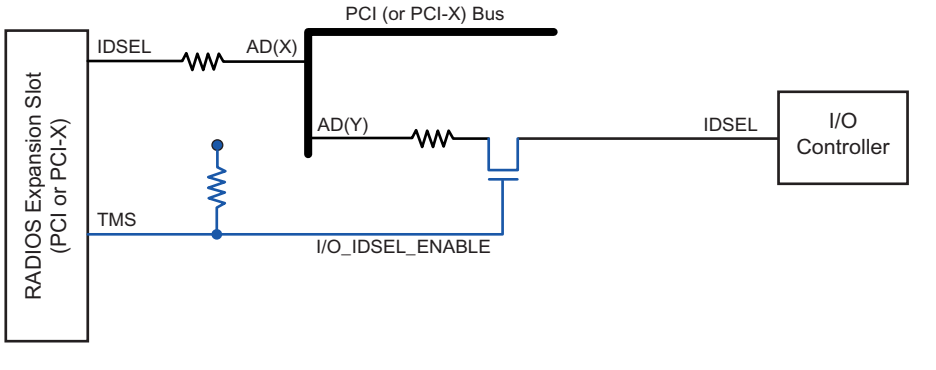
1. Insert a *MUX switch* in the AD line between the PCI (or PCI-X) bus and the IDSEL pin of the I/O controller used in the RAID subsystem (see Figure 5). Connect the input pin of the MUX to IDSEL of the I/O controller. Connect one output pin of the MUX to the AD line leading to the system bus. Connect the second output pin of the MUX to ground.

Note: A MUX (1 input, 2 output) is recommended in this implementation because it eliminates the need for a pull-down resistor on the IDSEL pin of the I/O controller. This removes the RC time constant and increases the speed of the switch when disconnecting the IDSEL signal from the bus. It also increases the compatibility of the RAIDIOS circuit with a wider range of MROMB (ZCR) adapters.

2. Connect a signal line from *TMS* on the RAIDIOS expansion slot to the control or enable input pin of the MUX..

A pull-up resistor is required on the TMS signal line to keep the signal from floating when the MROMB adapter is not inserted.

Figure 5. IDSEL Control Circuit Implementation

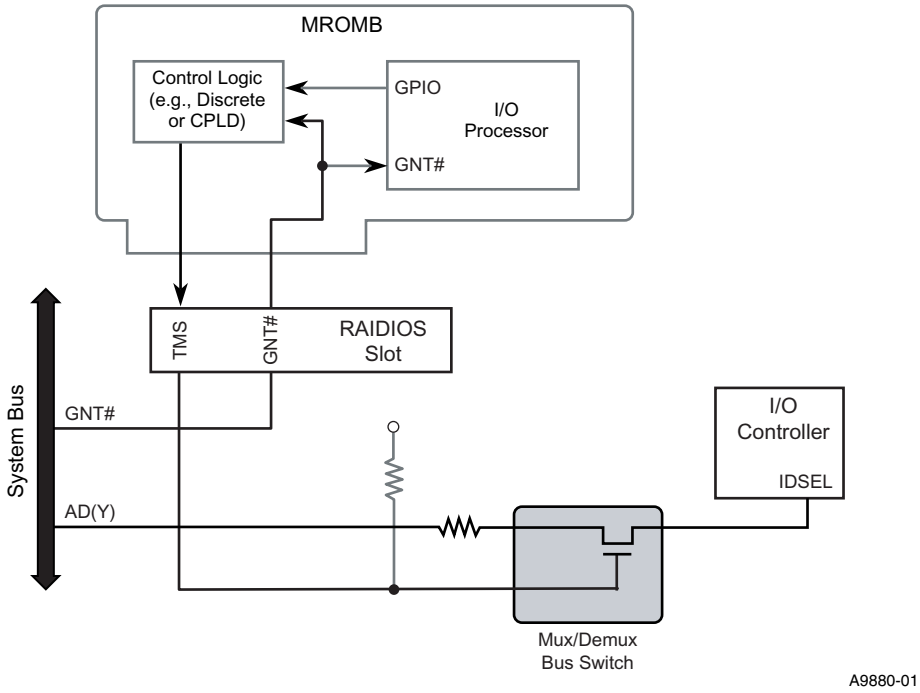


2.2.2 Theory of Operation

When the system is powered up with the MROMB adapter inserted, the TMS signal is driven low. This low signal is routed to the switch causing the switch to be disabled/opened. Opening the switch disconnects the IDSEL line of the I/O controller from the PCI (or PCI-X) bus, hiding it from the host system. When the MROMB adapter is ready to configure the I/O controller, it drives the TMS signal high making the IDSEL signal of the I/O controller visible on the system PCI (or PCI-X) bus. The MROMB adapter issues configuration cycles, configures the I/O controller, and requests system resource space from the host. Once the I/O controller is configured the MROMB adapter again drives the TMS signal low and hides the I/O controller from any possible host initiated configuration cycles. Care must be taken to ensure that IDSEL for the I/O controller is accessible to the system PCI (or PCI-X) bus *only* during the time that the MROMB adapter is configuring the device. The MROMB adapter accomplishes this by driving the TMS signal high *only* when the MROMB adapter has been given control of the bus (granted GNT#) by the bus arbiter. Figure 6 shows an example of how the MROMB adapter (with an Intel IOP) accomplishes this. Thus, in RAIDIOS systems, the implementation of the control logic is the responsibility of the MROMB adapter. This ensures that the host does not see the I/O controller during normal host PCI bus scans or when the MROMB is configuring the I/O controller.

Refer to Appendix A for a schematic view of the complete RAIDIOS circuit.

Figure 6. MROMB IDSEL Control Circuit Block Diagram Using an Intel IOP



2.3 Multiple I/O Controller Considerations

As S-ATA matures, MROMB designs that use S-ATA I/O controllers embedded on the motherboard for low cost RAID solutions will become very likely. S-ATA controllers initially will support only two to four hard drive connects (ports). For effective RAID solutions, multiple S-ATA controllers will be necessary. Thus the MROMB I/O processor will need to be able to configure and control multiple I/O controllers located on the motherboard.

RAIDIOS supports dynamic discovery of the I/O controller(s) in the same method that the host would use to configure each I/O controller. This allows MROMB adapters to discover and configure multiple I/O controllers located on the same system PCI (or PCI-X) bus segment. For multi-I/O controller MROMB applications there are several possible 'Wire-Or' configurations of the I/O controllers' interrupt lines for use with RAIDIOS. However, the RAIDIOS circuit itself does not require any modification. Each configuration can be optimal depending upon the I/O controllers' PCI device-types (i.e. single vs. multi-function), the number of I/O controllers, and the flexibility of the design (e.g. if supported by the MROMB firmware, is there a mechanism on the motherboard, such as a jumper selection, that allows for specific I/O controllers to be included or not included in the RAIDIOS-controlled RAID subsystem). The implementation example in this guide is based on dual I/O controllers (either single or multi-function PCI devices) where one could be selectable to be included or not be included in the RAID subsystem.

Note: The number of I/O controllers that can be enabled using the RAIDIOS specification is only limited by the PCI (or PCI-X) bus segment load limitations and the number of unique interrupts lines required per I/O controller. RAIDIOS, as currently designed, only supports the two interrupt lines INTC and INTD inbound to the RAIDIOS-enabled expansion slot. Most RAID firmware

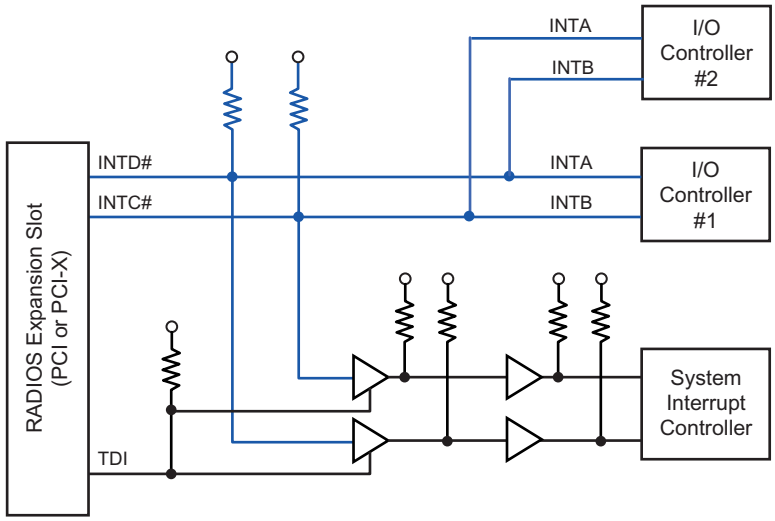
implementations will likely require that only I/O controllers of the same type/model be allowed to combine ('Wire-Or') interrupts.

2.3.1 Routing the Multiple Embedded I/O Controllers' Interrupts (INTx)

2.3.1.1 Example Implementation

For dual I/O controllers, 'Wire-Or' the interrupt signals of the two I/O controllers as shown in Figure 7 and follow the implementation procedures as explained in Section 2.1.1.

Figure 7. I/O Interrupt Routing Circuit for Multiple I/O Controllers



2.3.1.2 Theory of Operation

Once the interrupt lines of the two I/O controllers have been arranged in a 'Wire-Or' configuration, the theory of operation is the same as indicated in section 2.1.2. This particular 'Wire-Or' configuration (see Figure 7) is recommended because it offers an optimal interrupt sharing solution for the configuration shown. It offers a balanced distribution of the interrupts for two I/O controllers with two interrupt lines each. Other configurations may be more appropriate depending on the number of I/O controllers and the number of interrupt-lines needed per I/O controller.

2.3.2 Configuring and Controlling Multiple I/O Controllers

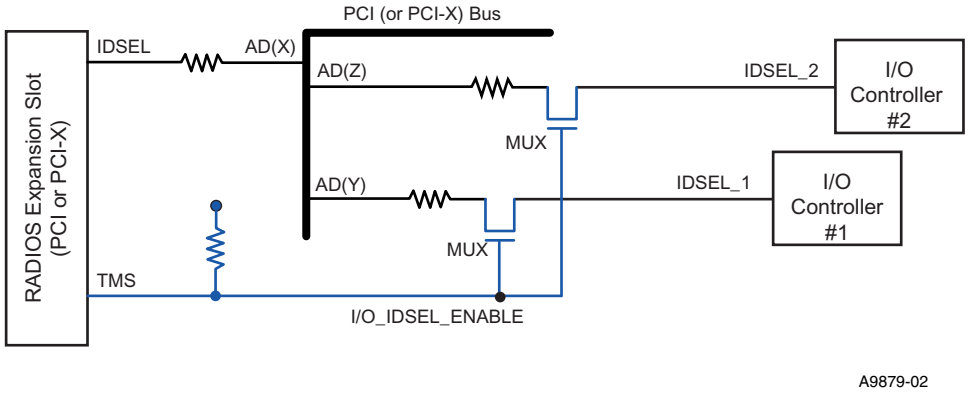
2.3.2.1 Implementation

Use the same implementation as indicated in Section 2.2.1 with the following additions. Run a branch from the TMS signal line to an additional MUX switch (see Figure 8). Route the IDSEL signal line of the second (or #2) I/O controller to one input of the MUX. Connect one output of the MUX to ground and the other output is connected to the system bus through the AD[z] line.

2.3.2.2 Theory of Operation

The theory of operation is the same for both single and multiple I/O controller solutions. Each I/O controller must have it's IDSEL signal connected to the system bus using a dedicated AD[x] line. This calls for an additional MUX (1 input, 2 output) switch for each I/O controller in the MROMB subsystem. The MUX is used in this circuit to eliminate the need for a pull-down resistor on the IDSEL input of each of the I/O controllers, thereby removing the RC time constant at the IDSEL input of the I/O controller. This then decreases the switching time of the switch and increases the robustness and compatibility of the RAIDIOS circuit.

Figure 8. IDSEL Control Circuit Implementation for Multiple I/O Controllers



2.4 Riser Cards With RAIDIOS

A riser card is a logical extension of the PCI (or PCI-X) expansion slot on the motherboard. A riser can be modified to extend the functionality of the RAIDIOS circuit to one of the riser card's expansion slots. Route the TMS and TDI signals as described in Section 2.1 and Section 2.2 to the expansion slot on the riser card that is RAIDIOS-enabled.

2.4.1 Implementation

1. Determine which expansion slot in the riser card will be the RAIDIOS expansion slot.
2. Connect the TMS and TDI signals for the riser card expansion slot to the connector pads of the bus interface connector on the riser card.

For all other expansion slots on the riser card follow normal design practices. When the riser card is inserted into the RAIDIOS expansion slot on the motherboard, the RAIDIOS circuit is passed through the motherboard's expansion slot and extended to the RAIDIOS expansion slot on the riser card.

2.5 I/O Bus Interface Requirements (PCI and PCI-X)

RAIDIOS is compatible with both PCI and PCI-X bus technologies. A key feature of the RAIDIOS design is its ability to provide system motherboard design engineers with one circuit that is compatible with both I/O buses. This simplifies both design and validation efforts for system motherboard designers.

RAIDIOS supports the maximum bus speed of both bus technologies (PCI 66 MHz and PCI-X 133 MHz). In a MROMB implementation, actual PCI-X bus speed is a determined by the system motherboard design and the MROMB adapter, not RAIDIOS.

3.0 RAIDIOS Design Specifications

3.1 Timing Requirements

The significant timing issue for MROMB or ROMB implementations is the amount of time from the assertion of *PCI Grant (GNT#)* to *IDSEL Valid.* (See Appendix B for PCI and PCI-X timing diagrams.) This amount of time includes the delay from *Clock* to *Grant Valid*, the propagation delay through the RAIDIOS logic to recognize Grant and enable the IDSEL switch, and the delay for the IDSEL switch to be enabled. This amount of time varies depending on the speed of the bus and the bus protocol (PCI vs. PCI-X). Table 2 and Table 3 detail worst case combinations (maximum timing limitations) of the circuits that are implemented on the motherboard.

For MROMB implementations, the timing budget for the RAIDIOS IDSEL control circuit on the motherboard is from the time *TMS Valid* is asserted by the MROMB adapter to *IDSEL Valid* for the I/O controller (the total propagation delay through the logic must be within the values specified in Table 2).

MROMB implementations contain the logic on the MROMB adapter to detect the assertion of *Grant (GNT#)* and to assert the TMS signal to the motherboard. ROMB implementations must include this logic on the motherboard and therefore the timing budget for ROMB is the timing budget of the MROMB implementation plus the timing budget of the MROMB adapter detecting *PCI Grant (GNT#)* to the time it asserts the *TMS Valid* signal to the motherboard (the total propagation delay through the logic must be within the values specified in Table 3).

Table 2. MROMB (ZCR): Motherboard Timing Budget From *TMS Valid* to *IDSEL Valid*

PROTOCOL	66 MHz	133 MHz
PCI	14 nanoseconds (ns)	
PCI-X		42ns

Table 3. ROMB: Motherboard Timing Budget From *GNT# Valid* to *IDSEL Valid*

PROTOCOL	66 MHz	133 MHz
PCI	19ns	
PCI-X		47ns

Note: The PCI-X timing budget is longer than the PCI timing budget because the PCI-X specification requires that the configuration cycle address phase be driven for four clock cycles before FRAME is valid. Also note (important for MROMB adapter design engineers) that the MROMB adapter's timing budget is 5 ns or less in order to meet the total timing budget requirement.

3.2 RAIDIOS Component List

Table 4 contains recommended RAIDIOS components. Ensure that when selecting parts all designs adhere to the timing specifications in Table 2 and Table 3.

Table 4. RAIDIOS Component List

Component	Qty (Single I/O controller)	Comments
IDSEL Control and Routing Circuit		
MUX/DEMUX Bus Switch	1	For example, the PI5C3303 Mux/Demux Bus Switch. It has a fast active-high enable and near zero propagation delay (typically < 0.25ns).
I/O Interrupt Control and Routing Circuit		
74125 Tri-state Buffer	2 (1 IC)	
7407 Hex Buffer	2 (1 IC)	High voltage, open collector outputs
Pull-up Resistors	9	4.7K ohm (typical)

4.0 MROMB Adapter Design Considerations for RAIDIOS

This section is intended as a high-level overview of design considerations that the MROMB adapter design engineer (to include RAID hardware and firmware) should consider when using RAIDIOS as the enabling specification for the MROMB solution. For more detailed development information when using Intel I/O processors in the MROMB adapter, visit www.developer.intel.com, where you will be able to find application notes regarding MROMB and RAIDIOS for the specific I/O processor being used.

4.1 Hiding and Configuring the Embedded I/O Controller

The MROMB adapter's firmware must hide the I/O controller from the host system. This is accomplished via the JTAG signal *TMS* using control logic on the MROMB adapter. The MROMB adapter must configure the I/O controller over a segment of the main system PCI (or PCI-X) bus. Therefore the MROMB adapter must un-hide the I/O controller to configure it. This could expose the I/O controller to the host. To eliminate the possibility of exposure during the MROMB initiated configuration cycles, the MROMB adapter must un-hide the I/O controller's IDSEL signal ***only*** when the MROMB IOP has been granted control of the bus by the bus arbiter. The MROMB adapter design engineer must include discrete logic that has as one of its inputs, the *GNT#* signal. See the schematic in the Appendix A for this implementation using an Intel IOP.

Note: The timing budget allowed for the MROMB adapter to route GNT# and assert TMS Valid is 5ns maximum. Refer to Table 2 and Table 3 for total timing requirements.

4.2 Routing the Embedded I/O Controller's Interrupts

The MROMB adapter must use the JTAG signal *TDI* to control the switches that route the I/O controller's interrupts. When the MROMB adapter is inserted in the RAIDIOS expansion slot, it must pull the TDI signal *low* to disable the tri-state buffers on the system motherboard. This leaves the branches to the RAIDIOS expansion slot as the only complete paths for the I/O controller's interrupt signals.

4.3 Impact to RAID Firmware of Multiple I/O Controllers

4.3.1 Hiding and Configuring Multiple I/O Controllers

There is not much difference between the multi-I/O controller case and the single I/O controller case. The RAID firmware must be able to dynamically discover the I/O controllers and assign a value to the BAR (Base Address Register) for each of the controllers' memory allocation space. All I/O controllers will be controlled together through the single TMS signal of the RAIDIOS expansion slot [i.e., I/O controllers cannot be hidden/unhidden individually].

4.3.2 Multiple I/O Controllers' Interrupt Routing Considerations

The RAID firmware will need to determine the proper I/O controller's interrupts to service. It's important to know the 'Wire-Or' configuration of the I/O controllers, as implemented by the motherboard design engineer.

5.0 Summary

RAIDIOS is a specification that Intel® developed for motherboards and zero-channel RAID cards. Design engineers can use RAIDIOS to implement MROMB solutions. RAIDIOS provides a simple, inexpensive, and versatile solution that can be used regardless of I/O bus technology (PCI or PCI-X).

At the time that this document was written, Intel had submitted and received Microsoft WHQL certification for one RAIDIOS-enabled platform. Thus RAIDIOS, as a specification for enabling motherboards to allow embedded I/O controllers used in MROMB applications, is accepted by Microsoft Corporation. Intel does not guarantee that a specific vendor's implementation will pass any specific certification program. Passing certification is the responsibility of each motherboard manufacturer.

Appendix A RAIDIOS Schematic

Figure 9. Single I/O Controller Example

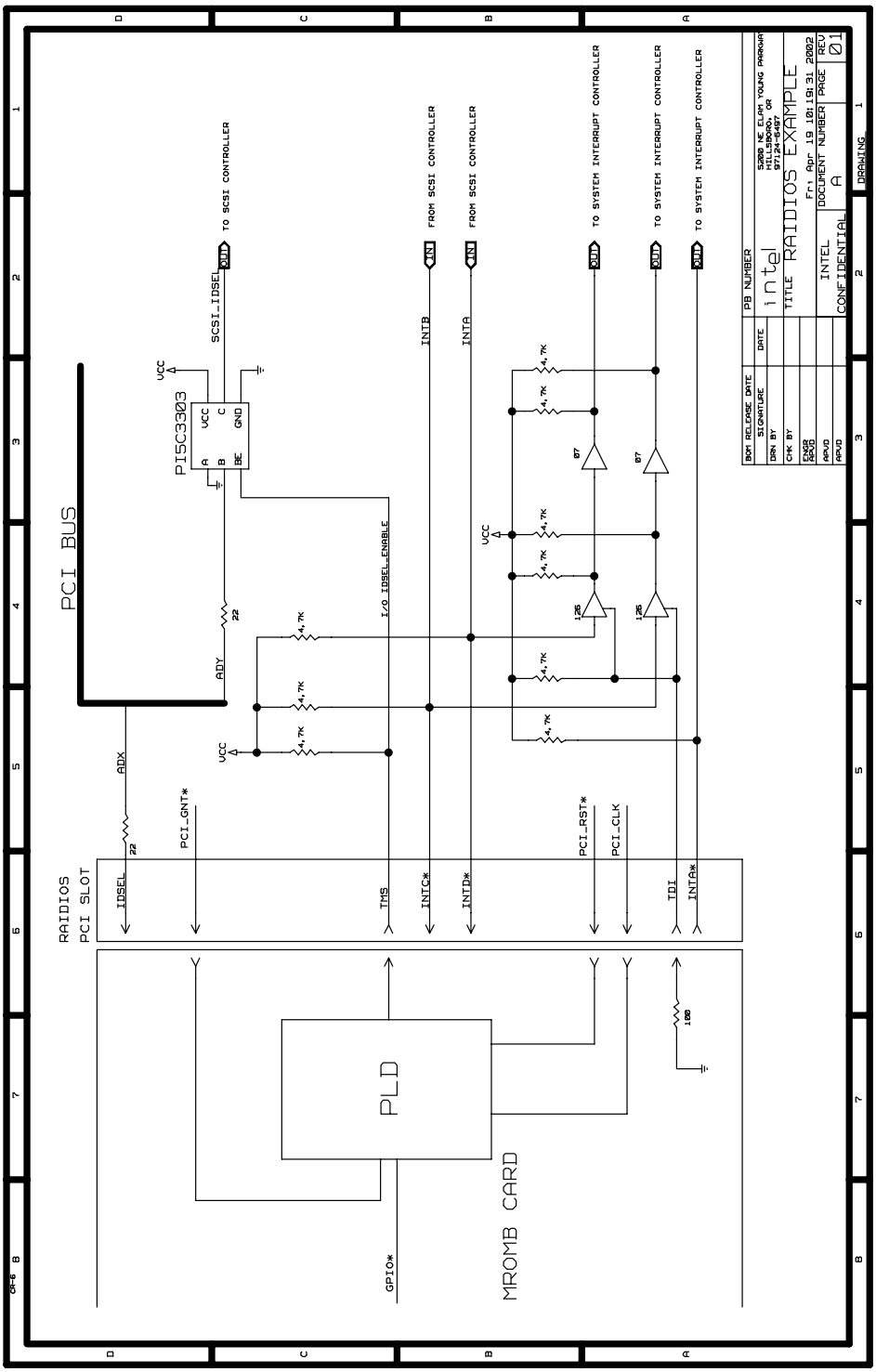
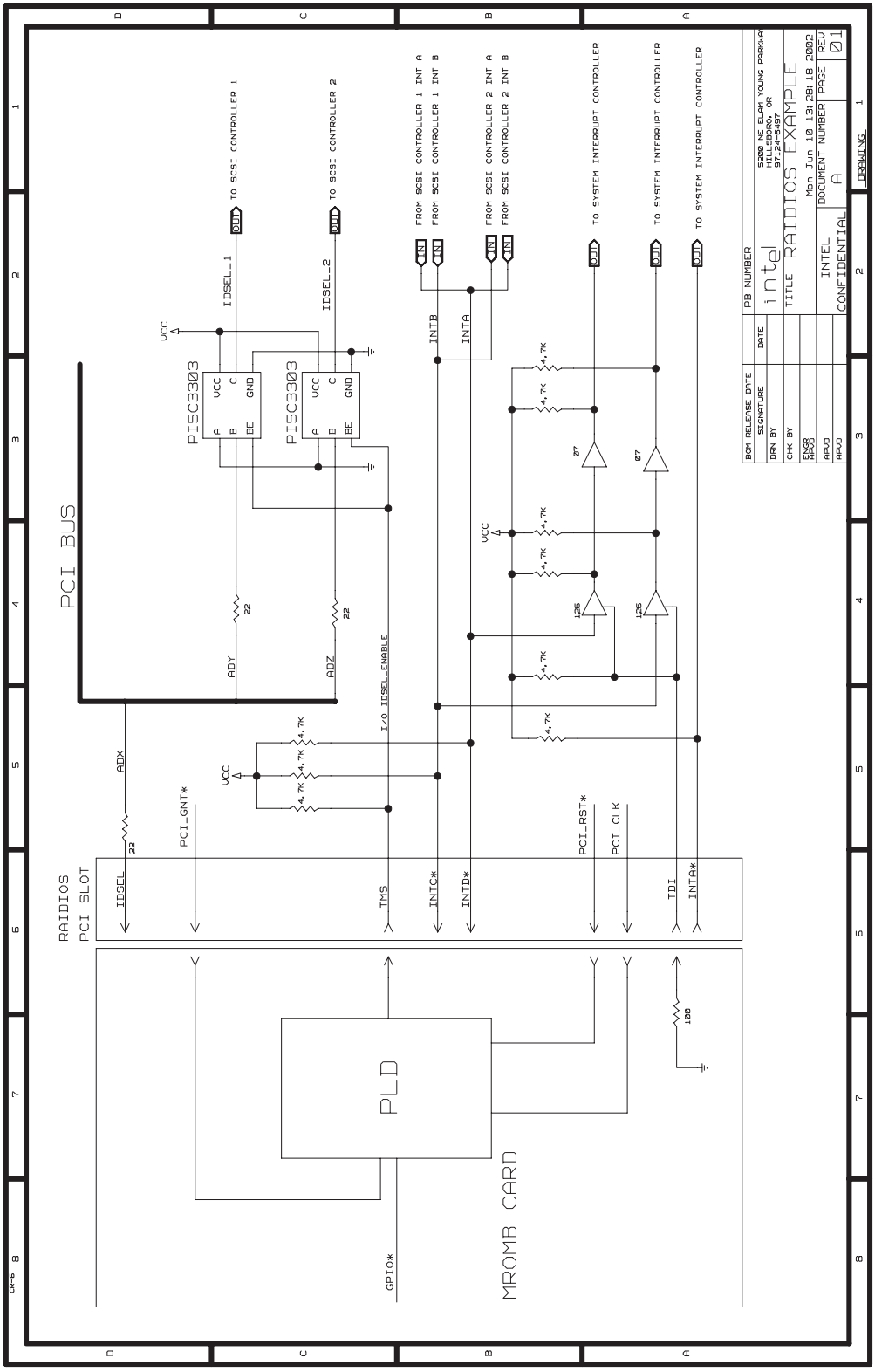


Figure 10. Multiple I/O Controller Example



Appendix B Timing Diagrams

Note: The following timing diagrams were generated from a specific configuration (with specific vendor parts) and are *examples only*. All designs should adhere to the maximum timing specifications as indicated in Table 2 and Table 3.

Figure 11. Example PCI Timing Diagram

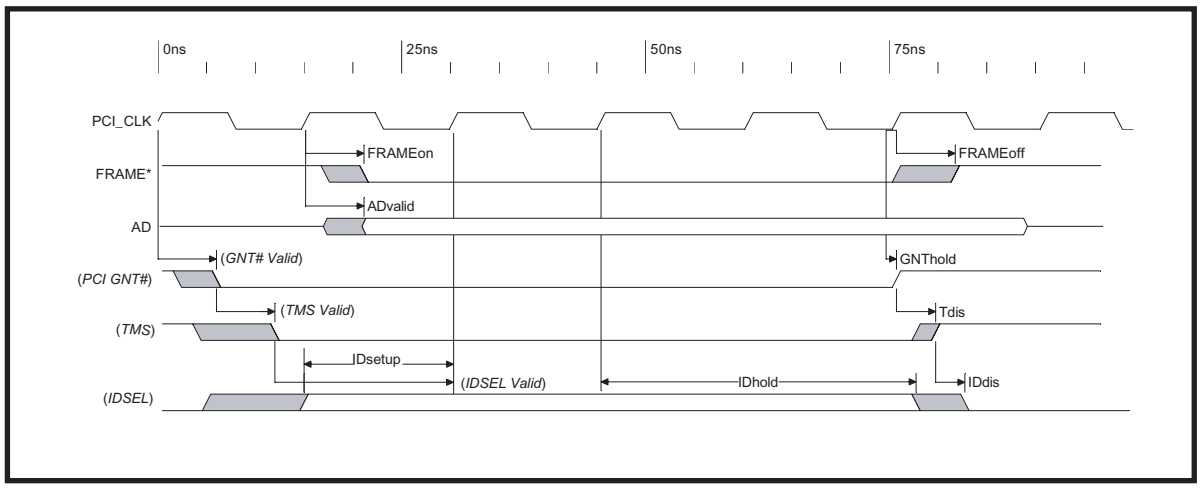


Figure 12. Example PCI-X Timing Diagrams